December 2002

INFORMATION QUALITY: MODELING ACCURACY- TIMELINESS IN TRANSACTION PROCESSING SYSTEMS

Narasimhaiah Gorla
Wayne State University

K. Krishnan
Wayne State University

INFORMATION QUALITY: MODELING ACCURACY-TIMELINESS IN TRANSACTION PROCESSING SYSTEMS

Narasimhaiah Gorla and K. S. Krishnan
Wayne State University
n_gorla@wayne.edu aa2365@wayne.edu

Abstract

Accurate and timely information is critical for managers in organizations in order to take right decision at the right time. In spite of the use of information technology in organizations for several decades, managers still do not receive accurate, timely, and relevant information. Errors in data cost companies millions or billions of dollars because of bad decisions and lost customers. In general accuracy / timeliness of information output in Transaction Processing Systems is affected by impurities in input data, time to correct data errors, the errors in the program, and time for program correction. In this article, we develop an analytical model of Information Accuracy-Timeliness as a function of input data errors and program errors. We provide managerial implications of this research for Information System (IS) development and IS resource allocation.

Motivation

Timely and accurate information is of essence for business managers who need to take the right decision at the right time. Managers cannot make right decisions if those decisions are based on out-dated or inaccurate information. In spite of the use of information technology in organizations for several decades, managers still do not receive accurate, timely and relevant information (Redman, 1995). Errors in data cost company millions or billions of dollars because of bad decisions and lost customers (Strong, Lee, and Wang, 1997; Redman, 1995). Thus, quality information is recognized as an important success factor for organizations.

Just like production systems, information systems gather data, process data into useful information and deliver information outputs to managers or other users (Wang, Storey, and Firth, 1995; Clikeman, 1999). A major reason for poor information quality is poor data quality. Based on previous case studies, it is expected that data error rates at field level ranges from 0.5% to 30%, but normally in the range of 1% to 5% (error rate = number of wrong fields/ number of total fields) (Redman, 1998). The impact of poor data quality at the operational level includes lower customer satisfaction and increased cost (typically 8-10% of revenue and for service organization 40-60% of expense). In addition, there are organizational impacts at the tactical level, such as, poor decision making, problems in data warehouse implementation, and organizational mistrust. The strategic level impacts stem from operational and tactical level impact. These include difficulty in setting and executing corporate strategy since these include data from external sources, which are more prone to errors (Redman, 1998).

The arrival of cheaper microcomputers and online terminals to mainframes has decentralization effect on information systems. The distributed information systems imply a large range of end users' participation in computer operations and a greater volume of information processed within the organization. As the end users with varied computer skills are involved in system operation, there is very likelihood of entering incorrect data by users of inadequate computer skills. Once these errors occur at the data entry level, these errors are caught and displayed by the edit program that handles data entry. These errors are then corrected by the data entry operator using the help menu feature of the edit program. This results in delay of producing computer outputs, thus affecting the timeliness of information. In transaction processing systems, most of time is consumed in preparation of input data to be accurate. In this paper, we model the effect of process of producing accurate information on timeliness of MIS outputs.

The research will be useful to IS managers as to decide the 'optimal' resource allocation between data preparation and software development. The research will also help determine the effect of user-interface quality on the timeliness of information outputs. Furthermore, a sensitivity analysis of the model the research will help the effect of end user computer skills on the accuracy and timeliness of information output. In addition, the relationship between software quality (reliability and maintainability) and the information quality (accuracy and timeliness) in the context of data preparation can be assessed.

The organization of this paper is as follows. In section 2, the characteristics of a transaction processing systems are described. In section 3, relevant previous research is discussed. Section 4 gives the development of initial model of information accuracy-timeliness tradeoff. Section 5 gives future extensions the authors are working on. Section 6 has summary and conclusions.

Transaction Processing System

A typical transaction processing system consists of collection of transaction data for an application, entering data into computer system, checking for data errors using a validation program as the data is entered, correcting the data errors using help menu, and generating information output using a processing program after the relevant transactions have been entered. Input data has to be prepared error free before it is sent to final processing. In transaction processing systems, most effort is consumed in preparing input data for processing: i.e. detecting input errors as completely and as accurately as possible and correcting those errors manually. Thus, the timeliness of information output is affected by the impurity of input data, the time consumed by users to correct impurities, and the inaccuracy of the edit program that checks for input errors. An erroneous program not only results in erroneous output but also results in processing delay because of the need for program correction. In this paper, we model the time of processing for producing accurate information output as a function of the accuracy of input data and correctness of computer program. We plan to provide some guidelines for MIS managers to follow in the system development process. We plan to recommend priorities for MIS managers for effective use of resources for obtaining quality information with the least delay.

Previous Related Research

Morey (1982) gives a functional relationship between three common measures of data quality: transaction reject rate, intrinsic transaction error rate and stored MIS record error rate. He also provides mechanisms to reduce the error rate in the MIS stored data. Ballou and Pazer (1985) identified four dimensions of data quality: accuracy, completeness, consistency, and timeliness. They presented a general model for multi-user information system wherein the percentage of output errors that are due to input errors and the percentage of output errors that are due to operational errors could be calculated. Ahituv and Zelek (1987) present a statistical method to monitor the quality of batch processing jobs by collecting and using historical data on consumption of hardware resources. Once the historical patterns are set, each batch run is compared with the historical data. Oman and Ayers (1988) discuss the information systems in the distributed environment. They present a methodology to improve data quality by using three phases: sending hard copy about percent errors, comparison of quality among organizations, and chief executive intervention. Ballou and Tayi (1989) used integer programming formulation to allocate resources committed to identifying and correcting data errors and proposed a heuristic to solve the problem. Ballou and Pazer (1995) model accuracy-timeliness tradeoff for situations wherein the information output becomes more accurate as the time passes by and more data is added. The tradeoff is the balance between preliminary but immediate information and accurate but delayed information. None of the previous research discusses the relationship between timeliness and accuracy of information output, as a function of data quality and program quality. In the study we develop an MIS Accuracy-Timeliness Model in terms of input data quality and program quality.

Development of Accuracy-Timeliness Model:

Loebbecke et. al (1983) classified errors and irregularities, by identifying the flow of transactions, into four categories: input errors (such as data lost, data inaccurate), processing errors (such as wrong file, incorrect processing), output errors (such as late or lost output) , and other errors (such as unlimited access, management override). In this paper, we consider input errors, processing errors and the resulting output errors. Figure 1 describes a simplified transaction processing system. In a transaction processing system, input data is entered into the system, which is validated by an EDIT program for errors. The function of EDIT program is to check each input data record and display the nature of data error explaining the reason for rejection. The data is corrected by the data entry operator using the help menu feature of the EDIT program. If the data error cannot be corrected, then the data is rejected and recorded for later checking. If the data is error free, the data is passed to PROCESS program for generating information output. The output of the PROCESS program is analyzed for any undetected errors. If the information output shows errors, both the EDIT program and the input data are checked and corrected as necessary. Then input is refed to EDIT program. The correctness of EDIT program is checked based on either the output of EDIT program and/ or output of PROCESS program. We make the following assumptions:

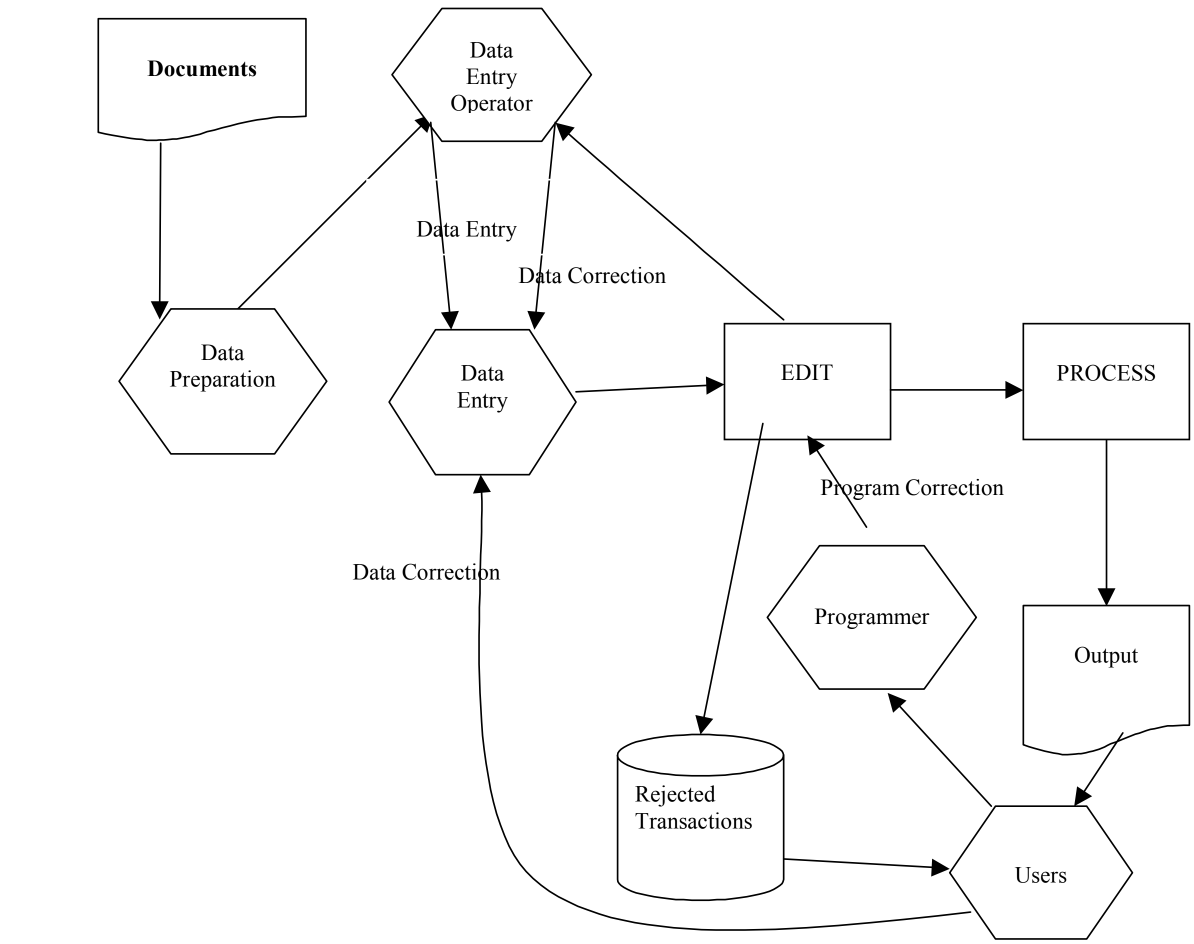

Figure 1. A Transaction Processing System

1. Input data may have errors; examples of such errors are invalid format, invalid data range, a corresponding master record being not available.

2. EDIT program may have logical errors. This means EDIT program may show correct input as incorrect and reject it or it may accept wrong input data as correct. Normally, if EDIT program does not report any input errors, the data is passed to PROCESS program.

3. PROCESS program has no logical errors. By observing the output from PROCESS program one can detect bad data that has passed through EDIT program undetected.

4. The corrective work is complete and accurate. i.e. if a input data or program is corrected for a specific error, there is no error in the correction. When the program is corrected for a specific error, it does not create another type of program error.

Model Derivation

Suppose that N transactions are waiting to be read into the system and edited for possible errors. If the editor is 100% reliable, then any error message given out by the editor can be attributed to errors in one or more fields in the input data. If the editor, however, is not 100% reliable, that is, if program errors exit in the editor, tracing the source of sources of errors can be very complicated. To facilitate presentation, we assume that if the source of error has been not identified in the input data, the editor program is reviewed for error and is debugged fully so that the editor program is error-free. We also assume that after all the N transactions have been edited, the data are processed using PROCESS program to generate a report. In order to estimate the expected total time for editing and generating reports, we consider seven cases as shown in Figure 2.

```
                                                           |--Detect the source     Case 1
                                                           |    Correctly
                    Error in                               |
                |-- Editor --- Detect Some Other Field--|--Detect some other     Case 2
                |   Program    as being  erroneous      | field as the source
                |                                       |
                |                                       |
                |                                       |--No error messages     Case 3
      |--Error--|
      | in data |
      |         |
      |         |--- No Error
      |              in Editor-----Detect the source
      |              Program       of error correctly---- Detect the source     Case 4
Input |                                                        correctly
Data ---|
      |                            |---Detect Error
      |              Error in      |   Wrongly                                  Case 5
      |          |--  Editor-----|
      |          |   Program       |
      |--No Erro |                 |---No Error Messages                        Case 6
        in data-|
                 |
                 |--- No Error
                      in Editor ----- No Error Messages                         Case 7
                      Program
```

Figure 2. Schematic Diagram for Various Cases of Data and Program Corrections

Let T_i denote the total expected time in Case i (i=1 to 7) from the moment an input data is read to the generation of report in the batch mode. (i=1 to 7). We define the following symbols:

t_{dc} = time to read and edit a single transaction
t_{dr} = time to correct errors in a single transaction if an error is detected
t_{er} = time to review and correct errors in the Edit program
t_p = time to process all edited transactions and generate a report.
p_1 = prob. of errors in one or more fields of a transaction
$1-p_1$ = prob. of no errors in the transaction
p_2 = prob. of program errors in the Editor
$1-p_2$ = prob. of no errors in the Editor
p_{31} = prob. (the editor detects errors in data correctly/ there is Editor error and there is data error)
p_{32} = prob. (the editor shows data errors incorrectly / there is Editor error and there is data error)
p_{33} = prob. (the editor detects no errors in data incorrectly/ there is Editor error and data error)
p_4 = prob. (the Editor detects errors in data correctly / there is Editor error and no data error)
$1-p_4$ = prob. (the Editor detects no errors in data incorrectly/there is Editor error and no data error)

Considering all the seven cases, the total expected time to generate a correct report =

$$(p_1p_2p_{31})T_1 + (p_1p_2p_{32})T_2 + (p_1p_2p_{33})T_3 + p_1(1-p_2)T_4 + p_1(1-p_2)p_4 + (1-p_1)p_2p_4\ T_5 + (1-p_1)p_2(1-p_4)T_6 + (1-p_1)(1-p_2)T_7$$

The coefficient of each of the T's is the probability of the corresponding case. For example, Case 1 involves detecting errors (p_1) in input data correctly (p_2) with errors in the Editor (p_{31}).

Summary

The model presented in this paper describes a two-stage system. It was assumed that the input data consists of N transactions waiting to be edited and entered into system one at a time. If the initial editing gives out an error message --rightly or wrongly—the source of error was identified and corrected. In case of a program error causing the editor to misread a faulty input record as being correct, the faulty record goes to the next stage undetected and the editor reads the next waiting input record.

When all the N transactions have been read and passed by the editor as being error-free, the N transactions are processed in the second stage for generating a final report. If a faulty input record escapes the scrutiny of the Editor in the first stage, we assumed that the second stage report generator (this includes both the report and the manual check of the report) would detect either data and/or program errors. The entire batch of N transactions is sent back for reading and editing each record once again as before.

Input data → [Enter Data] → [Editing of Data] → [Generating Report] → Final Report

One may also conceive of this model as a stochastic process (e.g, Markov Process) depicting the flow of data through edit and report generation process. It is important to note that the error in EDIT program can lead to erroneously misdiagnosing an error-free field in the input data as erroneous requiring corrective action on the EDIT program itself. Thus one may regard the data flow as a three dimensional Markov process where Input Data and Edit Program comprising two dimensions, and the two stages in the flow of data forming the third dimension. There are several ways of analyzing such a "solid" Markov process. We chose to break down the system into seven distinct cases for analyzing the underlying dynamics of the system. The seven cases describe the solid Markov process resulting in one of two trapping cases: Case 6 or Case 7. The use of solid Markov chain would also require analyzing several cases, hence we don't see inherent advantage of a solid Markov approach in the present research. We'll demonstrate the equivalence of the two approaches in our future research.

References

Will be provided when requested.